

DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDD	DDD	CCC	LLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL
DDDDDDDDDDDDDD		CCCCCCCCCCCC	LLLLLLLLLLLLLLLL

```
RRRRRRRR  PPPPPPPP  SSSSSSSS  UU      UU  BBBB BBBB
RRRRRRRR  PPPPPPPP  SSSSSSSS  UU      UU  BBBB BBBB
RR      RR  PP      PP  SS      SS  UU      UU  BB      BB
RR      RR  PP      PP  SS      SS  UU      UU  BB      BB
RR      RR  PP      PP  SS      SS  UU      UU  BB      BB
RRRRRRRR  PPPPPPPP  SSSSSS      UU      UU  BBBB BBBB
RRRRRRRR  PPPPPPPP  SSSSSS      UU      UU  BBBB BBBB
RR      RR  PP      PP      SS      SS  UU      UU  BB      BB
RR      RR  PP      PP      SS      SS  UU      UU  BB      BB
RR      RR  PP      PP      SS      SS  UU      UU  BB      BB
RR      RR  PP      PP      SSSSSSSS  UU      UU  BBBB BBBB
RR      RR  PP      PP      SSSSSSSS  UUUUUUUUU  BBBB BBBB
```

```
....
....
....
....
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```

(2)	54	DECLARATIONS
(3)	77	GET QUALIFIER DESCRIPTOR BLOCK
(4)	138	FIND COMMAND QUALIFIER
(5)	203	EXTRACT RESULT DESCRIPTOR FIELDS
(6)	265	SET RESULT DESCRIPTOR ADDRESS
(7)	296	GET PARAMETER
(8)	345	RESULT PARSE INIT

```
0000 1      .TITLE  RPSUB - DCL RESULT PARSE SUBROUTINES
0000 2      .IDENT  'V04-000'
0000 3
0000 4
0000 5      *****
0000 6
0000 7      COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      ALL RIGHTS RESERVED.
0000 10
0000 11      THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12      ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13      INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14      COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15      OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16      TRANSFERRED.
0000 17
0000 18      THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19      AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20      CORPORATION.
0000 21
0000 22      DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23      SOFTWARE OR EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25      *****
0000 26
0000 27
0000 28
0000 29      ++
0000 30      FACILITY:  STARLET DCL CLI
0000 31
0000 32      ABSTRACT:  MISC SUBROUTINES
0000 33
0000 34
0000 35      ENVIRONMENT:  NATIVE MODE USER CODE
0000 36
0000 37      AUTHOR:      W.H.BROWN, CREATION DATE:14-APR-77
0000 38
0000 39      MODIFIED BY:
0000 40
0000 41      V03-003 PCG0003      Peter George      15-Feb-1983
0000 42      Update to new structure level.
0000 43      Handle larger PTR data structure.
0000 44      Move DCL$CNVASCBIN to CONVERT.
0000 45
0000 46      V03-002 PCG0002      Peter George      14-Nov-1982
0000 47      Call DCL$TRIM to process the numeric string
0000 48      before converting it to an integer.
0000 49
0000 50      V03-001 PCG0001      Peter George      30-Sep-1982
0000 51      Use new larger PTR data structure.
0000 52      --
```

```

0000 54      .SBTTL  DECLARATIONS
0000 55      ::
0000 56      :: MACRO LIBRARY CALLS
0000 57      ::
0000 58
0000 59      PRCDEF      : DEFINE PROCESS WORK AREA
0000 60      WRKDEF      : DEFINE COMMAND WORK AREA
0000 61      $SCLITABDEF : DEFINE TABLE STRUCTURE
0000 62      PTRDEF      : DEFINE RESULT PARSE DESCRIPTOR
0000 63      RPWDEF      : RESULT PARSE WORK DEFINITIONS
0000 64      PLMDEF      : PARAMETER LIMIT DEFINITIONS
0000 65      $CLIDEF     : CLI DEFINITIONS
0000 66      $CLMSGDEF   : CLI MESSAGE DEFINITIONS
0000 67
0000 68
0000 69      ::
0000 70      :: OWN STORAGE:
0000 71      ::
00000000 72      .PSECT  DCL$ZCODE      BYTE, RD, NOWRT
0000 73

```

```
0000 75
0000 76      .DSABL  LSB
0000 77      .SBTTL  GET QUALIFIER DESCRIPTOR BLOCK
0000 78      :++
0000 79      : FUNCTIONAL DESCRIPTION:
0000 80
0000 81      : THIS ROUTINE IS CALLED TO LOCATE THE COMMAND QUALIFIER
0000 82      : DESCRIPTOR BLOCK FOR A SPECIFIC QUALIFIER.
0000 83      : ALTERNATE ENTRY TO CHECK THAT QUALIFIER IS A PARAMETER
0000 84      : QUALIFIER AS OPPOSED TO AN OUTPUT SPECIFIER.
0000 85
0000 86      : CALLING SEQUENCE:
0000 87
0000 88      : BSB/JSB DCL$GETQUALDESC      : GET QUALIFIER DESCRIPTOR
0000 89      : BSB/JSB DCL$GETPARMQUAL    : GET PARAMETER QUALIFIER DESCRIPTOR
0000 90
0000 91      : INPUT PARAMETERS:
0000 92
0000 93      : R1 IS THE CODE TO IDENTIFY THE QUALIFIER
0000 94
0000 95      : IMPLICIT INPUTS:
0000 96
0000 97      : R8 = ADDRESS OF UTILITY BIT ARRAY
0000 98      : R9 = ADDRESS OF REQUEST DESCRIPTOR
0000 99      : R10 = ADDRESS OF WORK BLOCK
0000 100     : R11 = ADDRESS OF PASS 1 PARSE WORK AREA
0000 101
0000 102     : OUTPUT PARAMETERS:
0000 103
0000 104     : R2 IS THE ADDRESS OF THE QUALIFIER DESCRIPTOR BLOCK
0000 105
0000 106     : COMPLETION CODES:
0000 107
0000 108     : R0 = SUCCESS/FAIL DEPENDING OF WHETHER THE DESCRIPTOR WAS FOUND
0000 109
0000 110     : SIDE EFFECTS:
0000 111
0000 112     : TOP LEVEL RETURN (RET) TAKEN IF SEARCH FAILS
0000 113
0000 114     :--
0000 115
0000 116     .ENABL  LSB
0000 117
0000 118     DCL$GETPARMQUAL::      : GET A PARAMETER QUALIFIER DESCRIPTOR
0000 119     DCL$GETQUALDESC::      : FIND A QUALIFIER DESCRIPTOR
0000 120     MOVL  R1,R0             : COPY QUALIFIER NUMBER
0000 121     BEQL  90$             : ZERO IS INVALID QUALIFIER NUMBER
0000 122     MOVL  WRK_L_QUABLK(R11),R2 : POINT AT START OF QUALIFIER BLOCKS
0000 123     BEQL  90$             : BR IF NONE
0000 124     BRB   20$            : START OF SEARCH
0000 125 10$: TSTL  ENT_L_NEXT(R2) : TEST OFFSET TO NEXT
0000 126     BEQL  90$             : BR IF THIS IS LAST
0000 127     ADDL3 ENT_L_NEXT(R2),- : FIND ADDRESS OF NEXT ENT BLOCK
0000 128     WRK_L_TAB_VEC(R11),R2
0000 129 20$: SOBGTR R0,10$      : COUNT DOWN QUALIFIER NUMBER
0000 130     INCL  R0               : INDICATE DESCRIPTOR FOUND
0000 131     RSB                  : BACK TO THE CALLER
```

50	51	D0	0000	120	MOVL	R1,R0	
	19	13	0003	121	BEQL	90\$	
52	CA	AB	D0	0005	122	MOVL	WRK_L_QUABLK(R11),R2
	13	13	0009	123	BEQL	90\$	
	0B	11	000B	124	BRB	20\$	
	0B	A2	D5	000D	125	10\$: TSTL	ENT_L_NEXT(R2)
	0C	13	0010	126	BEQL	90\$	
	0B	A2	C1	0012	127	ADDL3	ENT_L_NEXT(R2),-
52	DE	AB		128		WRK_L_TAB_VEC(R11),R2	
	F2	50	F5	0018	129	20\$: SOBGTR	R0,10\$
	50	D6	001B	130	INCL	R0	
		05	001D	131	RSB		

RPSUB
V04-000

- DCL RESULT PARSE SUBROUTINES
GET QUALIFIER DESCRIPTOR BLOCK

E 7

16-SEP-1984 00:14:19 VAX/VMS Macro V04-00
4-SEP-1984 23:43:05 [DCL.SRC]RPSUB.MAR;1

Page 4
(3)

04	001E	132			
	001E	133	90\$:	SETSTAT INVQUALNUM	: SET ERROR-INVALID QUALIFER NUMBER
	0023	134		RET	: GO BACK TO DISPATCHER
	0024	135			
	0024	136		.DSABL LSB	

```
0024 138 .SBTTL FIND COMMAND QUALIFIER
0024 139 :++
0024 140 : FUNCTIONAL DESCRIPTION:
0024 141 :
0024 142 : THIS COROUTINE IS CALLED TO SEARCH FOR A
0024 143 : COMMAND QUALIFIER IN THE RANGE OF THE CURRENT COMMAND.
0024 144 : THE SEARCH IS DONE OUT TO THE FIRST PARAMETER APPEARING
0024 145 : IN THE COMMAND, THEN FROM THE START OF THE FIRST PARAMETER
0024 146 : IN THE RANGE OF THE CURRENT COMMAND TO THE END OF THE
0024 147 : RANGE OF THE CURRENT COMMAND.
0024 148 :
0024 149 : CALLING SEQUENCE:
0024 150 :
0024 151 : BSB/JSB DCL$LOCMDQUAL
0024 152 :
0024 153 : INPUT PARAMETERS:
0024 154 :
0024 155 : R1 IS THE CODE OF THE QUALIFIER TO LOCATE
0024 156 :
0024 157 : IMPLICIT INPUTS:
0024 158 :
0024 159 : R8 = ADDRESS OF UTILITY BIT ARRAY
0024 160 : R9 = ADDRESS OF REQUEST DESCRIPTOR
0024 161 : R10 = ADDRESS OF WORK BLOCK
0024 162 : R11 = ADDRESS OF PASS 1 PARSE WORK AREA
0024 163 :
0024 164 : OUTPUT PARAMETERS:
0024 165 :
0024 166 : R4 IS RETURNED AS THE ADDRESS OF THE DESCRIPTOR IF FOUND
0024 167 : R5 IS THE INDEX TO THE DESCRIPTOR IF FOUND
0024 168 :
0024 169 : COMPLETION CODES:
0024 170 :
0024 171 : R0 IS SET TRUE OR FALSE DEPENDING OF SUCCESS OF SEARCH
0024 172 :
0024 173 : SIDE EFFECTS:
0024 174 :
0024 175 : REGISTERS R4, R5 & R6 ARE USED BY THIS ROUTINE
0024 176 : AND MUST BE PRESERVED ACCROSS COROUTINE RETURNS.
0024 177 :
0024 178 :--
0024 179 :
0024 180 DCL$FNDCMDQUAL::
0024 181 : LOCATE THE COMMAND QUALIFIER
56 55 01 D0 0024 181 MOVL #1,R5 : SET INDEX TO START SEARCH
0024 182 : START OF PARAMETER LIMIT DESCRIPTORS
0024 183 10$: MOVAL RPW_G_PRMLIN(R10),R6 :
0024 184 : ASSUME NO MORE QUALIFIERS
0024 185 : THIS DESCRIPTOR IN RANGE OF VERB
0024 186 : BR IF YES
0024 187 20$: CMPB R5,RPW_B_STRPARM(R10) : BR IF WITHIN A PARAMETER
0024 188 : SET INDEX OF PLACE TO START LOOKING
0024 189 30$: BLSSU 40$ : BR WHEN DONE
0024 190 : IS THIS WITHIN THE CURRENT PARAMETER?
0024 191 40$: BNEQ 30$ : BR IF OUT OF RANGE OF THIS PARAMETER
0024 192 : SET ADDRESS OF RESULT DESCRIPTOR
0024 193 : YIELD LIMITS FOR TYPE
0024 194 : IF THIS A COMMAND QUALIFIER?
0024 195 : BR IF NO-CONTINUE SEARCH
```

56 55 01 D0 0024 181 MOVL #1,R5 : LOCATE THE COMMAND QUALIFIER
56 40 AA DE 0027 182 : SET INDEX TO START SEARCH
0024 183 10\$: MOVAL RPW_G_PRMLIN(R10),R6 : START OF PARAMETER LIMIT DESCRIPTORS
08 AA 55 91 0028 184 : ASSUME NO MORE QUALIFIERS
0024 185 : THIS DESCRIPTOR IN RANGE OF VERB
0024 186 : BR IF YES
55 01 A6 9A 0031 187 20\$: CMPB R5,RPW_B_STRPARM(R10) : BR IF WITHIN A PARAMETER
0024 188 : SET INDEX OF PLACE TO START LOOKING
0024 189 30\$: BLSSU 40\$: BR WHEN DONE
02 A6 55 91 0033 190 : IS THIS WITHIN THE CURRENT PARAMETER?
0024 191 40\$: BNEQ 30\$: BR IF OUT OF RANGE OF THIS PARAMETER
0024 192 : SET ADDRESS OF RESULT DESCRIPTOR
0024 193 : YIELD LIMITS FOR TYPE
0024 194 : IF THIS A COMMAND QUALIFIER?
0024 195 : BR IF NO-CONTINUE SEARCH

50	D6	004A	195	INCL	R0	:	SET SUCCESS
9E	16	004C	196	JSB	@(SP)+	:	RETURN WITH QUALIFIER
55	D6	004E	197 50\$:	INCL	R5	:	ADVANCE INDEX TO NEXT DESCRIPTOR
D9	11	0050	198	BRB	10\$:	CHECK AGAIN
56	04	C0	0052 199 60\$:	ADDL	#PLM_K_SIZE,R6	:	SET TO NEXT PARAMETER LIMIT DESCRIPTOR
DE	11	0055	200	BRB	20\$:	TRY NEXT PARAMETER
	05	0057	201 70\$:	RSB		:	RETURN WITH VALUE OR ZERO

```
0058 203 .SBTTL EXTRACT RESULT DESCRIPTOR FIELDS
0058 204 :++
0058 205 : FUNCTIONAL DESCRIPTION:
0058 206 :
0058 207 : THIS ROUTINE IS CALLED TO TAKE A RESULT DESCRIPTOR APART
0058 208 : AND RETURN ITS COMPONENT PART AS INDIVIDUAL VALUES.
0058 209 :
0058 210 : CALLING SEQUENCE:
0058 211 :
0058 212 : BSB/JSB DCL$EXTNXTDESC : EXTRACT NEXT DESCRIPTOR
0058 213 : BSB/JSB DCL$GETEXTDESC : GET AND EXTRACT DESCRIPTOR
0058 214 : BSB/JSB DCL$EXTRSLDESC : EXTRACT RESULT DESCRIPTOR
0058 215 :
0058 216 : INPUT PARAMETERS:
0058 217 :
0058 218 : AT EXTRSLDESC WITH R4 CONTAINS THE ADDRESS OF THE DESCRIPTOR
0058 219 : AT EXTNXTDESC WITH R6 CONTAINS THE ADDRESS OF THE PARAMETER
0058 220 : LIMIT DESCRIPTOR.
0058 221 :
0058 222 : IMPLICIT INPUTS:
0058 223 :
0058 224 : R8 = ADDRESS OF UTILITY BIT ARRAY
0058 225 : R9 = ADDRESS OF REQUEST DESCRIPTOR
0058 226 : R10 = ADDRESS OF WORK BLOCK
0058 227 : R11 = ADDRESS OF PASS 1 PARSE WORK AREA
0058 228 :
0058 229 : OUTPUT PARAMETERS:
0058 230 :
0058 231 : R1 = TYPE
0058 232 : R2 = SIZE OR VALUE DEPENDING ON THE DESCRIPTOR
0058 233 : R3 = ADDRESS OF THE ITEM
0058 234 : R4 = ADDRESS OF DESCRIPTOR
0058 235 :
0058 236 : COMPLETION CODES:
0058 237 :
0058 238 : R0 = SUCCESS/FAILURE DEPENDING ON RESULT OF SEARCH
0058 239 :
0058 240 :--
0058 241 : .ENABL LSB
0058 242 :
0058 243 DCL$EXTNXTDESC:: : EXTRACT NEXT COMPLETE DESCRIPTOR
0058 244 : SETSTAT FAIL : ASSUME WONT FIND ONE
0058 245 : MOVZBL PLM_B_NXTDESC(R6),R5 : SET POINTER TO DESCRIPTOR
0058 246 : BEQL 40$ : BR IF PARAMETER SET IS MISSING
0058 247 : CMPB R5,PLM_B_LSTDESC(R6) : IS THIS IN RANGE OF CURRENT PARAMETER?
0058 248 : BGTRU 40$ : BR IF NO
0058 249 : INCB PLM_B_NXTDESC(R6) : ADVANCE INDEX TO NEXT
0058 250 : INCL R0 : SET ANY SUCCESSFUL STATUS
0058 251 DCL$GETEXTDESC:: : GET AND EXTRACT NEXT DESCRIPTOR
0058 252 : BSB B DCL$SETDESCADR : SET ADDRESS OF RESULT DESCRIPTOR
0058 253 DCL$EXTRSLDESC:: : EXTRACT FOR RESULT DESCRIPTION
0058 254 : EXTZV #PTR_V_OFFSET,#PTR_S_OFFSET,- : START BIT AND SIZE OF OFFSET
0058 255 : PTR_C_DESCR(R4),R3 : GET OFFSET INTO R3
0058 256 : MOVAB WRK_G_BUFFER(R11)[R3],R3 : FIND ADDRESS OF ITEM IN BUFFER
0058 257 : EXTZV #PTR_V_VALUE,#PTR_S_VALUE,- : START BIT AND SIZE OF VALUE
0058 258 : PTR_C_DESCR(R4),R2 : GET VALUE INTO R2
0058 259 : EXTZV #PTR_V_TYPE,#PTR_S_TYPE,- : START BIT AND SIZE OF TYPE
```

55	66	9A	005A	245	MOVZBL	PLM_B_NXTDESC(R6),R5
02	A6	55	005D	246	BEQL	40\$
		1B	005F	247	CMPB	R5,PLM_B_LSTDESC(R6)
		66	0063	248	BGTRU	40\$
		50	0065	249	INCB	PLM_B_NXTDESC(R6)
			0067	250	INCL	R0
		16	0069	251	DCL\$GETEXTDESC::	
			0069	252	BSB B	DCL\$SETDESCADR
			006B	253	DCL\$EXTRSLDESC::	
	0C	08	006B	254	EXTZV	#PTR_V_OFFSET,#PTR_S_OFFSET,-
	53	64	006E	255		PTR_C_DESCR(R4),R3
53	F492	CB43	0070	256	MOVAB	WRK_G_BUFFER(R11)[R3],R3
	0B	00	0076	257	EXTZV	#PTR_V_VALUE,#PTR_S_VALUE,-
	52	64	0079	258		PTR_C_DESCR(R4),R2
	04	1C	007B	259	EXTZV	#PTR_V_TYPE,#PTR_S_TYPE,-

RPSUB
V04-000

- DCL RESULT PARSE SUBROUTINES
EXTRACT RESULT DESCRIPTOR FIELDS

I 7

16-SEP-1984 00:14:19 VAX/VMS Macro V04-00
4-SEP-1984 23:43:05 [DCL.SRC]RPSUB.MAR;1

Page 8
(5)

```
51 64 05 007E 260 PTR_L_DESCR(R4),R1 ; GET TYPE INTO R1
      0080 261 408: RSB ;
      0081 262
      0081 263 .DSABL LSB
```

```
0081 265 .SBTTL SET RESULT DESCRIPTOR ADDRESS
0081 266 :++
0081 267 FUNCTIONAL DESCRIPTION:
0081 268
0081 269 THIS ROUTINE IS CALLED TO SET THE ADDRESS OF A RESULT
0081 270 DESCRIPTOR INTO R4.
0081 271
0081 272 CALLING SEQUENCE:
0081 273
0081 274 BSB/JSB DCL$SETDESCADR
0081 275
0081 276 INPUT PARAMETERS:
0081 277
0081 278 R5 CONTAINS THE INDEX FOR THE DESIRED DESCRIPTOR
0081 279
0081 280 IMPLICIT INPUTS:
0081 281
0081 282 R10 = ADDRESS OF WORK BLOCK
0081 283 R11 = ADDRESS OF PASS 1 PARSE WORK AREA
0081 284
0081 285 OUTPUT PARAMETERS:
0081 286
0081 287 R4 IS LOADED WITH THE ADDRESS OF THE DESCRIPTOR
0081 288
0081 289 :--
0081 290
0081 291 DCL$SETDESCADR::
0081 292 MULL3 #PTR_C_LENGTH,R5,R4 : GET BYTE OFFSET OF DESCRIPTOR
54 55 0C C5 0081 293 MOVAB WRK_G_RESULT-PTR_C_LENGTH(R11)[R4],R4 : GET ADDRESS OF DESCRIPTOR
54 F9AA CB44 9E 0085 294 RSB
05 008B
```

```
008C 296 .SBTTL GET PARAMETER
008C 297 **
008C 298 FUNCTIONAL DESCRIPTION:
008C 299
008C 300 THIS ROUTINE IS CALLED TO SEARCH THE RESULT DESCRIPTOR
008C 301 BUFFER FOR THE NEXT OCCURANCE OF A PRARMETER
008C 302
008C 303 CALLING SEQUENCE:
008C 304
008C 305 BSB/JSB DCL$GETPARM ; GET A PARAMETER
008C 306
008C 307 INPUT PARAMETERS:
008C 308
008C 309 R5 CONTAINS THE INDEX OF NEXT DESCRIPTOR TO CHECK
008C 310
008C 311 IMPLICIT INPUTS:
008C 312
008C 313 R8 = ADDRESS OF UTILITY BIT ARRAY
008C 314 R9 = ADDRESS OF REQUEST DESCRIPTOR
008C 315 R10 = ADDRESS OF WORK BLOCK
008C 316 R11 = ADDRESS OF PASS 1 PARSE WORK AREA
008C 317
008C 318 OUTPUT PARAMETERS:
008C 319
008C 320 R1 CONTAINS THE TYPE OF THE DESCRIPTOR(IE:PTR_K_PARAMETR)
008C 321 R2 CONTAINS THE SIZE OF THE PARAMETER
008C 322 R3 CONTAINS THE PRECEEDING TERMINATOR
008C 323 R4 CONTAINS THE ADDRESS OF THE PARAMETER DESCRIPTOR
008C 324 R5 IS THE INDEX TO THE DESCRIPTOR
008C 325
008C 326 COMPLETION CODES:
008C 327
008C 328 R0 = SUCCESS/FAIL DEPENDING ON THE RESULT OF THE SEARCH
008C 329
008C 330 --
008C 331
008C 332 DCL$GETPARM::
008C 333 SETSTAT FAIL ; GET THE NEXT PARAMETER
008C 334 10$: INCL R5 ; ASSUME NO MORE PARAMETERS
008C 335 BSBB DCL$GETTEXTDESC ; ADVANCE INDEX
008C 336 EXTZV #PTR_V_TERM,#PTR_S_TERM, - ; GET DESCRIPTOR AND EXTRACT FIELDS
008C 337 -PTR_C_LENGTH(R4),R3 ; GET THE TERMINATOR FORM THE PRVIOUS
008C 338 CMPB R1,#PTR_K_ENDLINE ; DESCRIPTOR AND SAVE IN R3
008C 339 BEQL 30$ ; IS THIS THE END OF LINE?
008C 340 CMPB R1,#PTR_K_PARAMETR ; NO MORE PARAMETERS
008C 341 BNEQ 10$ ; IS THE CURRENT A PARAMETER?
008C 342 SETSTAT SUCCESS ; OR IF NO-TRY NEXT
008C 343 30$: RSB ; SET FOUND ONE
; RETURN TO CALLER
```

53	04	F4	A4	91	0098	338
	04		51	91	009B	339
	03		08	13	009D	340
			51	91	00A0	341
			EC	12	00A2	342
				05	00A5	343

```
00A6 345 .SBTTL RESULT PARSE INIT
00A6 346
00A6 347 **
00A6 348 FUNCTIONAL DESCRIPTION:
00A6 349
00A6 350 THIS ROUTINE IS CALLED TO ESTABLISH INITIAL CONDITIONS
00A6 351 IN THE RESULT PARSE WORK AREA PRIOR TO PERFORMAING
00A6 352 A RESULT PARSE.
00A6 353
00A6 354 CALLING SEQUENCE:
00A6 355
00A6 356 ENTERED VIA A CASE FOLLOWED BY A CALL
00A6 357
00A6 358 IMPLICIT INPUTS:
00A6 359
00A6 360 R9 = ADDRESS OF REQUEST DESCRIPTOR
00A6 361 R10 = ADDRESS OF WORK BLOCK
00A6 362 R11 = ADDRESS OF PASS 1 PARSE WORK AREA
00A6 363
00A6 364 OUTPUT PARAMETERS:
00A6 365
00A6 366 THE RESULT PARSE WORK AREA IS INITED
00A6 367
00A6 368 COMPLETION CODES:
00A6 369
00A6 370 R0 = SUCCESS
00A6 371
00A6 372
00A6 373 DCLSRPINIT::
00A6 374
00A6 375 BSBB DCL$GETDCLWRK
00A6 376 MOVCS #0,(R10),#0,-
00A6 377 #CLISC WORKAREA-4,4(R10)
00A6 378
00A6 379 MOVL R11,RPWL DCLWRK(R10)
00A6 380 MOVL WRK_L_PROPTR(R11),R7
00A6 381 MOVAL RPWL_G_PRMLIM(R10),R6
00A6 382 CLRL R5
00A6 383 BSBB DCL$GETPARM
00A6 384 MOVB R5,RPWL_B_STRPAM(R10)
00A6 385 BLBC R0,90$
00A6 386 10$: MOVB R5,PLM_B_FSTDESC(R6)
00A6 387 20$: MOVB R5,PLM_B_NXTDESC(R6)
00A6 388 BSBB DCL$GETPARM
00A6 389 SUBB3 #1,R5,PLM_B_LSTDESC(R6)
00A6 390 BLBC R0,90$
00A6 391 CMPB R3,#PTR_K_BLANK
00A6 392 BEQL 50$
00A6 393 BBS #ENT V IMPCAT,-
00A6 394 ENT B FLAGS(R7),20$
00A6 395 CMPB R3,#PTR_K_COMMA
00A6 396 BNEQ 20$
00A6 397 40$: CMPB R3,#PTR_K_BLANK
00A6 398 BEQL 50$
00A6 399 BSBB DCL$GETPARM
00A6 400 BLBC R0,90$
00A6 401 BRB 40$
00A6 402 50$: ADDL3 ENT_L_NEXT(R7),-
00A6 403 WRK_L_TAB_VEC(R11),R7
00A6 404
00A6 405
00A6 406
00A6 407
00A6 408
00A6 409
00A6 410
00A6 411
00A6 412
00A6 413
00A6 414
00A6 415
00A6 416
00A6 417
00A6 418
00A6 419
00A6 420
00A6 421
00A6 422
00A6 423
00A6 424
00A6 425
00A6 426
00A6 427
00A6 428
00A6 429
00A6 430
00A6 431
00A6 432
00A6 433
00A6 434
00A6 435
00A6 436
00A6 437
00A6 438
00A6 439
00A6 440
00A6 441
00A6 442
00A6 443
00A6 444
00A6 445
00A6 446
00A6 447
00A6 448
00A6 449
00A6 450
00A6 451
00A6 452
00A6 453
00A6 454
00A6 455
00A6 456
00A6 457
00A6 458
00A6 459
00A6 460
00A6 461
00A6 462
00A6 463
00A6 464
00A6 465
00A6 466
00A6 467
00A6 468
00A6 469
00A6 470
00A6 471
00A6 472
00A6 473
00A6 474
00A6 475
00A6 476
00A6 477
00A6 478
00A6 479
00A6 480
00A6 481
00A6 482
00A6 483
00A6 484
00A6 485
00A6 486
00A6 487
00A6 488
00A6 489
00A6 490
00A6 491
00A6 492
00A6 493
00A6 494
00A6 495
00A6 496
00A6 497
00A6 498
00A6 499
00A6 500
00A6 501
00A6 502
00A6 503
00A6 504
00A6 505
00A6 506
00A6 507
00A6 508
00A6 509
00A6 510
00A6 511
00A6 512
00A6 513
00A6 514
00A6 515
00A6 516
00A6 517
00A6 518
00A6 519
00A6 520
00A6 521
00A6 522
00A6 523
00A6 524
00A6 525
00A6 526
00A6 527
00A6 528
00A6 529
00A6 530
00A6 531
00A6 532
00A6 533
00A6 534
00A6 535
00A6 536
00A6 537
00A6 538
00A6 539
00A6 540
00A6 541
00A6 542
00A6 543
00A6 544
00A6 545
00A6 546
00A6 547
00A6 548
00A6 549
00A6 550
00A6 551
00A6 552
00A6 553
00A6 554
00A6 555
00A6 556
00A6 557
00A6 558
00A6 559
00A6 560
00A6 561
00A6 562
00A6 563
00A6 564
00A6 565
00A6 566
00A6 567
00A6 568
00A6 569
00A6 570
00A6 571
00A6 572
00A6 573
00A6 574
00A6 575
00A6 576
00A6 577
00A6 578
00A6 579
00A6 580
00A6 581
00A6 582
00A6 583
00A6 584
00A6 585
00A6 586
00A6 587
00A6 588
00A6 589
00A6 590
00A6 591
00A6 592
00A6 593
00A6 594
00A6 595
00A6 596
00A6 597
00A6 598
00A6 599
00A6 600
00A6 601
00A6 602
00A6 603
00A6 604
00A6 605
00A6 606
00A6 607
00A6 608
00A6 609
00A6 610
00A6 611
00A6 612
00A6 613
00A6 614
00A6 615
00A6 616
00A6 617
00A6 618
00A6 619
00A6 620
00A6 621
00A6 622
00A6 623
00A6 624
00A6 625
00A6 626
00A6 627
00A6 628
00A6 629
00A6 630
00A6 631
00A6 632
00A6 633
00A6 634
00A6 635
00A6 636
00A6 637
00A6 638
00A6 639
00A6 640
00A6 641
00A6 642
00A6 643
00A6 644
00A6 645
00A6 646
00A6 647
00A6 648
00A6 649
00A6 650
00A6 651
00A6 652
00A6 653
00A6 654
00A6 655
00A6 656
00A6 657
00A6 658
00A6 659
00A6 660
00A6 661
00A6 662
00A6 663
00A6 664
00A6 665
00A6 666
00A6 667
00A6 668
00A6 669
00A6 670
00A6 671
00A6 672
00A6 673
00A6 674
00A6 675
00A6 676
00A6 677
00A6 678
00A6 679
00A6 680
00A6 681
00A6 682
00A6 683
00A6 684
00A6 685
00A6 686
00A6 687
00A6 688
00A6 689
00A6 690
00A6 691
00A6 692
00A6 693
00A6 694
00A6 695
00A6 696
00A6 697
00A6 698
00A6 699
00A6 700
00A6 701
00A6 702
00A6 703
00A6 704
00A6 705
00A6 706
00A6 707
00A6 708
00A6 709
00A6 710
00A6 711
00A6 712
00A6 713
00A6 714
00A6 715
00A6 716
00A6 717
00A6 718
00A6 719
00A6 720
00A6 721
00A6 722
00A6 723
00A6 724
00A6 725
00A6 726
00A6 727
00A6 728
00A6 729
00A6 730
00A6 731
00A6 732
00A6 733
00A6 734
00A6 735
00A6 736
00A6 737
00A6 738
00A6 739
00A6 740
00A6 741
00A6 742
00A6 743
00A6 744
00A6 745
00A6 746
00A6 747
00A6 748
00A6 749
00A6 750
00A6 751
00A6 752
00A6 753
00A6 754
00A6 755
00A6 756
00A6 757
00A6 758
00A6 759
00A6 760
00A6 761
00A6 762
00A6 763
00A6 764
00A6 765
00A6 766
00A6 767
00A6 768
00A6 769
00A6 770
00A6 771
00A6 772
00A6 773
00A6 774
00A6 775
00A6 776
00A6 777
00A6 778
00A6 779
00A6 780
00A6 781
00A6 782
00A6 783
00A6 784
00A6 785
00A6 786
00A6 787
00A6 788
00A6 789
00A6 790
00A6 791
00A6 792
00A6 793
00A6 794
00A6 795
00A6 796
00A6 797
00A6 798
00A6 799
00A6 800
00A6 801
00A6 802
00A6 803
00A6 804
00A6 805
00A6 806
00A6 807
00A6 808
00A6 809
00A6 810
00A6 811
00A6 812
00A6 813
00A6 814
00A6 815
00A6 816
00A6 817
00A6 818
00A6 819
00A6 820
00A6 821
00A6 822
00A6 823
00A6 824
00A6 825
00A6 826
00A6 827
00A6 828
00A6 829
00A6 830
00A6 831
00A6 832
00A6 833
00A6 834
00A6 835
00A6 836
00A6 837
00A6 838
00A6 839
00A6 840
00A6 841
00A6 842
00A6 843
00A6 844
00A6 845
00A6 846
00A6 847
00A6 848
00A6 849
00A6 850
00A6 851
00A6 852
00A6 853
00A6 854
00A6 855
00A6 856
00A6 857
00A6 858
00A6 859
00A6 860
00A6 861
00A6 862
00A6 863
00A6 864
00A6 865
00A6 866
00A6 867
00A6 868
00A6 869
00A6 870
00A6 871
00A6 872
00A6 873
00A6 874
00A6 875
00A6 876
00A6 877
00A6 878
00A6 879
00A6 880
00A6 881
00A6 882
00A6 883
00A6 884
00A6 885
00A6 886
00A6 887
00A6 888
00A6 889
00A6 890
00A6 891
00A6 892
00A6 893
00A6 894
00A6 895
00A6 896
00A6 897
00A6 898
00A6 899
00A6 900
00A6 901
00A6 902
00A6 903
00A6 904
00A6 905
00A6 906
00A6 907
00A6 908
00A6 909
00A6 910
00A6 911
00A6 912
00A6 913
00A6 914
00A6 915
00A6 916
00A6 917
00A6 918
00A6 919
00A6 920
00A6 921
00A6 922
00A6 923
00A6 924
00A6 925
00A6 926
00A6 927
00A6 928
00A6 929
00A6 930
00A6 931
00A6 932
00A6 933
00A6 934
00A6 935
00A6 936
00A6 937
00A6 938
00A6 939
00A6 940
00A6 941
00A6 942
00A6 943
00A6 944
00A6 945
00A6 946
00A6 947
00A6 948
00A6 949
00A6 950
00A6 951
00A6 952
00A6 953
00A6 954
00A6 955
00A6 956
00A6 957
00A6 958
00A6 959
00A6 960
00A6 961
00A6 962
00A6 963
00A6 964
00A6 965
00A6 966
00A6 967
00A6 968
00A6 969
00A6 970
00A6 971
00A6 972
00A6 973
00A6 974
00A6 975
00A6 976
00A6 977
00A6 978
00A6 979
00A6 980
00A6 981
00A6 982
00A6 983
00A6 984
00A6 985
00A6 986
00A6 987
00A6 988
00A6 989
00A6 990
00A6 991
00A6 992
00A6 993
00A6 994
00A6 995
00A6 996
00A6 997
00A6 998
00A6 999
00A6 1000
```

Address	Hex	Op	Label	Comment
03 A6	55	90	00FA 402	MOVB R5,PLM_B_TRMDESC(R6) ; SAVE DESCRIPTOR OF PARAMETER TERMINATOR
	86	D5	00FE 403	TSTL (R6)+ ; POINT AT NEXT PARAMETER LIMIT DESCRIPTOR
	C6	11	0100 404	BRB 10\$; SCAN NEXT PARAMETER
			0102 405	90\$: SETSTAT SUCCESS ; SET ALL IS GOOD
03 A6	55	90	0105 406	MOVB R5,PLM_B_TRMDESC(R6) ; SAVE FINAL TERMINATOR
		04	0109 407	RET ; RETURN TO DISPATCHER
			010A 408	
			010A 409	:: SET WORK ADDRESS
			010A 410	
			010A 411	DCL\$GETDCLWRK:: ;
00000000'EF	16		010A 412	JSB CLI\$GET_PRC ; GET ADDRESS OF CLI PROCESS WORK AREA
5B 04 AB	D0		0110 413	MOVL PRC_L_SAVFP(R11),R11 ; GET ADDRESS OF COMMAND WORK AREA
	05		0114 414	RSB ; RETURN TO CALLER
			0115 415	END

CLISC WORKAREA	= 00000080			PRC_L_OUTRABCTX	00000118
CLISGET_PRC	*****	X	02	PRC_L_PPFLIST	00000070
CLIS_INVQUALNUM	= 0003881A			PRC_L_RECALLPTR	0000012F
DCLSEXTNXTDESC	00000058	RG	02	PRC_L_RESTART	00000058
DCLSEXTSLDESC	00000068	RG	02	PRC_L_SAVAP	00000000
DCLSFNDCMDQUAL	00000024	RG	02	PRC_L_SAVFP	00000004
DCLSGETDCLWRK	0000010A	RG	02	PRC_L_SEVERITY	00000050
DCLSGETEXTDESC	00000069	RG	02	PRC_L_SPWN	000000C0
DCLSGETPARM	0000008C	RG	02	PRC_L_STACKLM	000000A4
DCLSGETPARMQUAL	00000000	RG	02	PRC_L_STACKPT	000000A0
DCLSGETQUALDESC	00000000	RG	02	PRC_L_STATUS	00000054
DCLSRPINIT	000000A6	RG	02	PRC_L_STS	00000084
DCLSETDESCADR	00000081	RG	02	PRC_L_STV	00000088
ENT_L_NEXT	= 00000008			PRC_L_SYMBOL	00000060
ENT_V_IMPCAT	= 00000007			PRC_L_TMBX	00000074
ENT_W_FLAGS	= 00000004			PRC_L_TRMLIST	00000010
PLM_B_FSTDESC	00000001			PRC_Q_ALLOCREG	00000020
PLM_B_LSTDESC	00000002			PRC_Q_COMMAND	000000E0
PLM_B_NXTDESC	00000000			PRC_Q_FLUSHTIME	000000D0
PLM_B_QUADDESC	00000003			PRC_Q_GLOBAL	00000028
PLM_B_TRMDESC	00000003			PRC_Q_IMAGENAME	000000D8
PLM_C_SIZE	00000004			PRC_Q_KEYPAD	00000040
PLM_K_SIZE	00000004			PRC_Q_LABEL	00000030
PRC_B_CONTINUE	000000F3			PRC_Q_LOCAL	00000038
PRC_B_DEFRADIX	000000AE			PRC_Q_SAVEPRIV	000000E8
PRC_B_EXMDEPMOD	000000AD			PRC_T_OUTDVI	0000011C
PRC_B_EXMDEPWID	000000AC			PRC_W_ASTIOSB	000000C6
PRC_B_EXONLYL	0000012D			PRC_W_ASTRETN	000000C8
PRC_B_FLAGS2	000000AF			PRC_W_ASTSTATUS	000000C4
PRC_B_IMGFLAG	00000078			PRC_W_ATTMBX	0000007A
PRC_B_OUTFLAGS	0000012C			PRC_W_FLAGS	00000068
PRC_B_PROMPTLEN	000000F0			PRC_W_INPCHAN	00000064
PRC_C_LENGTH	00000534			PRC_W_ONLEVEL	0000006A
PRC_G_COMMANDS	00000133			PRC_W_OUTIFI	00000114
PRC_G_PROMPT	000000F4			PRC_W_OUTISI	00000116
PRC_K_LENGTH	00000534			PRC_W_OUTMBXCHN	000000CA
PRC_L_CURRKEY	00000048			PRC_W_OUTMBXREF	000000CE
PRC_L_EXMDEPADR	000000A8			PRC_W_OUTMBXSIZ	000000CC
PRC_L_EXTARG	00000094			PRC_W_PMPCTRL	000000F1
PRC_L_EXTBLK	0000008C			PRC_W_WAITIOSB	00000066
PRC_L_EXTCOD	0000009C			PTR_B_LEVEL	00000004
PRC_L_EXTHND	00000090			PTR_B_NUMBER	00000005
PRC_L_EXTPRM	00000098			PTR_B_PARMCNT	00000006
PRC_L_IDFLNK	000000BC			PTR_B_VALUE	00000000
PRC_L_IMGACTSTS	00000080			PTR_C_LENGTH	0000000C
PRC_L_INDCLOCK	0000007C			PTR_K_BLANK	= 00000001
PRC_L_INDEPTH	0000005C			PTR_K_CMDQUAL	= 00000000
PRC_L_INDFAB	0000001C			PTR_K_COMMA	= 00000005
PRC_L_INDINPRAB	00000014			PTR_K_ENDLINE	= 00000004
PRC_L_INDOUTRAB	00000018			PTR_K_LENGTH	0000000C
PRC_L_INPRAB	00000008			PTR_K_PARAMETR	= 00000003
PRC_L_LASTKEY	0000004C			PTR_L_DESCR	00000000
PRC_L_LSTSTATUS	000000B0			PTR_L_ENTITY	00000008
PRC_L_ONCTLY	000000B8			PTR_S_OFFSET	= 0000000C
PRC_L_ONERROR	0000006C			PTR_S_TERM	= 00000004
PRC_L_OUTOFBAND	000000B4			PTR_S_TYPE	= 00000004
PRC_L_OUTRAB	0000000C			PTR_S_VALUE	= 00000008

PTR_V_OFFSET	= 00000008
PTR_V_TERM	= 00000018
PTR_V_TYPE	= 0000001C
PTR_V_VALUE	= 00000000
RPW_B_LSTDESC	00000009
RPW_B_STRPARM	00000008
RPW_C_HDRSIZ	00000040
RPW_C_LENGTH	00000080
RPW_G_BITS	00000020
RPW_G_PRMLIM	00000040
RPW_K_HDRSIZ	00000040
RPW_K_LENGTH	00000080
RPW_L_DCLWRK	00000004
RPW_L_USERCTX	00000000
WRK_B_CMDOPT	FFFFFFFFC3
WRK_B_MAXPARM	FFFFFFFFD0
WRK_B_MINPARM	FFFFFFFFD1
WRK_B_PARMCNT	FFFFFFFFCE
WRK_B_PARMSUM	FFFFFFFFCF
WRK_B_RECALLCNT	FFFFFFFFC5
WRK_B_VALLEV	FFFFFFFFC4
WRK_B_VERBTYP	FFFFFFFFC2
WRK_C_LENGTH	FFFFFF486
WRK_G_BUFFER	FFFFFF492
WRK_G_INPBUF	FFFFFF896
WRK_G_RESULT	FFFFFF9B6
WRK_K_LENGTH	FFFFFF486
WRK_L_CHARPTR	FFFFFF48E
WRK_L_DISALLOW	FFFFFFFE6
WRK_L_ERRORRTN	FFFFFF9AE
WRK_L_EXPANDPTR	FFFFFF486
WRK_L_IMAGE	FFFFFFFE2
WRK_L_MARKPTR	FFFFFF48A
WRK_L_PAROUT	FFFFFFFD2
WRK_L_PMPTADDR	FFFFFF9A2
WRK_L_PROMPTRTN	FFFFFF9A6
WRK_L_PROPTR	FFFFFFFC6
WRK_L_QUABLK	FFFFFFFCA
WRK_L_READRTN	FFFFFF9AA
WRK_L_RECALLPTR	FFFFFFFEA
WRK_L_RSLEND	FFFFFFFB6
WRK_L_RSLNXT	FFFFFFFBA
WRK_L_SAVAP	FFFFFFF8
WRK_L_SAVFP	FFFFFFFC
WRK_L_SAVSP	FFFFFFF4
WRK_L_SIGNALRTN	FFFFFFFD6
WRK_L_SPECRTN	FFFFFF9B2
WRK_L_TAB_VEC	FFFFFFDE
WRK_L_VERB	FFFFFFBE
WRK_W_FLAGS	FFFFFFF0
WRK_W_FLAGS2	FFFFFFF2
WRK_W_IMGCHAN	FFFFFFEE
WRK_W_PMPTLEN	FFFFFF99E

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	FFFFFFFFC (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
DCL\$ZCODE	00000115 (277.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	9	00:00:00.11	00:00:00.42
Command processing	83	00:00:00.81	00:00:05.05
Pass 1	250	00:00:09.13	00:00:22.27
Symbol table sort	0	00:00:01.20	00:00:02.81
Pass 2	72	00:00:01.69	00:00:06.40
Symbol table output	20	00:00:00.16	00:00:00.68
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	436	00:00:13.14	00:00:37.67

The working set limit was 1200 pages.
45492 bytes (89 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 815 non-local and 18 local symbols.
415 source lines were read in Pass 1, producing 13 object records in Pass 2.
34 pages of virtual memory were used to define 20 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
\$255\$DUA28:[SYSLIB]SYSBLDMLB.MLB;1	0
\$255\$DUA28:[DCL.OBJ]DCL.MLB;1	8
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	12

956 GETS were required to define 12 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:RPSUB/OBJ=OBJ\$:RPSUB MSRC\$:RPSUB/UPDATE=(ENH\$:RPSUB)+EXECML\$/LIB+LIB\$:DCL/LIB+SYSS\$LIBRARY:SYSBLDMLB/LIB

0073 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

